The following security alert was issued by the Information Security Division of the Mississippi Department of ITS and is intended for State government entities. The information may or may not be applicable to the general public and accordingly, the State does not warrant its use for any specific purposes.

TLP: WHITE
Disclosure is not limited. Subject to standard copyright rules, TLP: WHITE information may be distributed without restriction.
http://www.us-cert.gov/tlp/

DATE(S) ISSUED:
10/25/2019

SUBJECT:
Multiple Vulnerabilities in PHP Could Allow for Arbitrary Code Execution

OVERVIEW:
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow for arbitrary code execution. PHP is a programming language originally designed for use in web-based applications with HTML content. PHP supports a wide variety of platforms and is used by numerous web-based software applications. Successfully exploiting the most severe of these vulnerabilities could allow for arbitrary code execution in the context of the affected application. Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights. Failed exploitation could result in a denial-of-service condition.

THREAT INTELLIGENCE:
There are currently no reports of these vulnerabilities being exploited in the wild.

SYSTEMS AFFECTED:
- PHP 7.1 Prior to Version 7.1.33
- PHP 7.2 Prior to Version 7.2.24
- PHP 7.3 Prior to Version 7.3.11

RISK:
Government:
- Large and medium government entities: **High**
- Small government: **High**

Businesses:
- Large and medium business entities: **High**
- Small business entities: **High**

Home users: Low

TECHNICAL SUMMARY:
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow for arbitrary code execution. Details of these vulnerabilities are as below:

Version 7.1.33
- Bug #78599 (env_path_info underflow in fpm_main.c can lead to RCE)

Version 7.2.24

- Bug #78535 (auto_detect_line_endings value not parsed as bool)
- Bug #78620 (Out of memory error)
- Bug #78442 ('Illegal component' on exif_read_data since PHP7)
- Bug #78599 (env_path_info underflow in fpm_main.c can lead to RCE)
- Bug #78579 (mb_decode_numericentity: args number inconsistency)
- Bug #78609 (mb_check_encoding()
- Bug #76809 (SSL settings aren't respected when persistent connections are used)
- Bug #78623 (Regression caused by "SP call yields additional empty result set")
- Bug #78624 (session_gc return value for user defined session handlers)
- Bug #76342 (file_get_contents waits twice specified timeout)
- Bug #78612 (strtr leaks memory when integer keys are used and the subject string shorter)
- Bug #76859 (stream_get_line skips data if used with data-generating filter)
- Bug #78641 (addGlob can modify given remove_path value)

Version 7.3.11

- Bug #78535 (auto_detect_line_endings value not parsed as bool)
- Bug #78620 (Out of memory error)
- Bug #78442 ('Illegal component' on exif_read_data since PHP7)
- Bug #78599 (env_path_info underflow in fpm_main.c can lead to RCE)
- Bug #78413 (request_terminate_timeout does not take effect after fastcgi_finish_request)
- Bug #78633 (Heap buffer overflow (read)
- Bug #78579 (mb_decode_numericentity: args number inconsistency)
- Bug #78609 (mb_check_encoding()
- Bug #76809 (SSL settings aren't respected when persistent connections are used)
- Bug #78525 (Memory leak in pdo when reusing native prepared statements)
- Bug #78272 (calling preg_match()
- Bug #78623 (Regression caused by "SP call yields additional empty result set")
- Bug #78624 (session_gc return value for user defined session handlers)
- Bug #76342 (file_get_contents waits twice specified timeout)
- Bug #78612 (strtr leaks memory when integer keys are used and the subject string shorter)
- Bug #76859 (stream_get_line skips data if used with data-generating filter)
- Bug #78641 (addGlob can modify given remove_path value)

Successfully exploiting the most severe of these vulnerabilities could allow for arbitrary code execution in the context of the affected application. Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights. Failed exploitation could result in a denial-of-service condition.

RECOMMENDATIONS:

The following actions should be taken:

- Upgrade to the latest version of PHP immediately, after appropriate testing.
- Verify no unauthorized system modifications have occurred on system before applying patch.
- Apply the principle of Least Privilege to all systems and services.

- Remind users not to visit websites or follow links provided by unknown or untrusted sources.

REFERENCES:

PHP:
https://www.php.net/ChangeLog-7.php#7.1.33
https://www.php.net/ChangeLog-7.php#7.2.24
https://www.php.net/ChangeLog-7.php#7.3.11

TLP: WHITE
Disclosure is not limited. Subject to standard copyright rules, TLP: WHITE information may be distributed without restriction.
http://www.us-cert.gov/tlp/

Chris Watts
Security Operations Analyst
MS Department of Information Technology Services
601-432-8201 | www.its.ms.gov